REMARKS

Applicants reply to the Final Office Action dated July 27, 2006, within two months. Thus, Applicants request an Advisory Action, if necessary. Claims 36-48 were pending in the application and the Examiner rejects claims 36-48. Support for the amendments may be found in the originally-filed specification, claims, and figures. No new matter has been introduced by these amendments. Applicants assert that the application is in condition for allowance and reconsideration of the pending claims is requested.

Objection to Drawings

The Examiner objects to the drawings as failing to comply with 37 CFR § 1.84(p)(4), "because reference character '188' has been used to designate both business unit class, key object class, key class, highest level key class and generically as a key in a database" (page 5, paragraph 3). Applicants respectfully disagree.

As stated below in reference to the 35 U.S.C. § 112 rejection, an object-oriented database stores the representation of data within class objects. Figure 7 illustrates a hierarchy, wherein a key class "188" is established as the highest level of the hierarchy of object classes. However, this is but one embodiment and the specification is clear in describing, for example, that the key class may be a "business unit class." The key (top level) class may be representative of any number of varying types of object classes, thus the present application makes reference to this key class in the context of it being at the top of a hierarchical class structure. In other words, the key 188 class is clearly referenced as being the top level class, and thus one of ordinary skill in the art would appreciate that a top level class may represent any number of class types, or be described in any number of ways. For example, depending on the context in which it is being described, a top level class ("key 188") may comprise a high level key class (as there may be more than one key class) or a business unit class (describing the utility of the class). Therefore, Applicants assert that the variances presented in reference to the top level "key" class are appropriate in that they are each a different embodiment of a key 188 class.

Rejection under 35 U.S.C. § 112

The Examiner rejects claims 42 and 43 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Specifically, the Examiner asserts that the, "claims refer to 'key object classes', 'secondary object classes' and that the 'key object classes' partition the database in accordance with high-level category" (page 6, paragraph 4). The Examiner asserts that the term, "key

object classes" is indefinite because a "key field" is known to be a record index in databases terminology. Applicants respectfully traverse the rejection.

Applicants respectfully assert that those skilled in the art would immediately appreciate that in terms of object-oriented database (object database) architecture, referencing an object or object class as a database field type is not only common, but appropriate. In an object database, information is represented as objects, and as in object-oriented programming, classes define the core functionality for the objects they will spawn. As such, data is stored as objects that can be manipulated using the methods defined within the class to which the object belongs. As in the relational database architecture, an object database may contain any number of tables that are linked together through indexes. Though the mechanics behind the linkages between the two database architectures vary significantly (an object database uses pointers as the linkage mechanism), it is not considered improper to use similar terminology to refer to indexes and key fields when describing the structure of an object database. Thus, because data is represented as objects within an object database, a "key" field can be synonymous with a class or a class of objects.

The disclosure of the instant application describes the relationship between a "key" field and an object class as follows; "Database 142 preferably contains a 'key' field that partitions the database according to a high-level class of objects. An example of a 'key' field is the 'business unit' class 188 shown in Figure 7" (page 15, paragraph 2). It would be apparent to one of ordinary skill that the "business unit" class is defined as the "key" field that partitions the database.

Rejection under 35 U.S.C. § 103(a)

The Examiner next rejects claims 36-48 under 35 U.S.C. § 103(a) as being unpatentable over Schein et al., U.S. Patent No. 6,226,623 ("Schein"). Applicants respectfully traverse these rejections.

In the Response to Arguments, the Examiner admits that, "Schein does not disclose his invention in terms of object-oriented paradigm" (page 4, paragraph 4). However, the Examiner next asserts that, "Owens was introduced to address applicant's concern over the absence of the terms class and object in Schein." Importantly, Applicants note that despite the Examiner's admission that Owens is needed, the 35 U.S.C. § 103(a) rejection has been maintained based on Schein only.

Contrary to the Examiner's suggestion, Applicants concerns about Schein are much more significant than Schein's lack of the terms "class" and "object." Schein not only lacks disclosure of these elements, but Shein also does not disclose the unique related processes performed by class objects.

Significantly, the Schein disclosure states that while, "a single central repository for storing all customer related information throughout a business offers significant potential, the database is necessarily so large that certain problems arise. For example, the present inventors recognize that a database of this size cannot practically be directly searched" (col. 11, lines 11-19). To help solve the problem of impractical searching, Schein describes a means to allow users to build programs for searching the central database. As such, the data management of Schein requires an additional layer of complexity in order to render the central relational database searchable in a reasonable manner.

In other words, relational databases are much more complex than object databases, so searching the relational database of Shein is much more complex because (i) the user needs to formulate more complex search algorithms to search the database; and (ii) the numerous linkages between tables requires more intensive processing of the numerous links. **While Schein attempts to alleviate the complexity from the user standpoint (as set forth in (i) above), Schein does not disclose or suggest how to alleviate the requirement for intensive processing.**

More specifically, a relational database is an efficient and practical tool for managing data under many circumstances. However, those of ordinary skill would immediately recognize that the efficiency of a relational database decreases in proportion to the complexity of the table structure and the volume of data maintained within. As tables are added to a relational database, linkages are added in order to tie records from the table to records in any number of other tables. These linkages are processor intensive and can become very complex in large databases. Schein discloses a large and complex relational database; therefore, Schein also recognizes the need to implement separate programming tools to reduce processor load and thus speed database searches. Object databases where introduced to alleviate these problems, in that class objects maintain the linkages through pointers, which is far less processor intensive.

As such, Schein does not disclose or suggest at least a, "second subsection containing a high-level secondary class of objects and a second plurality of secondary classes of objects derived from said high-level secondary class of objects, wherein each of said second plurality of secondary classes of objects define one of said plurality of stored value products; and, wherein said second plurality of secondary classes of objects inherit attributes from said high-level key class of objects," as recited by independent claim 36.

Claims 37-48 depend from independent claim 36. Dependent claims 37-48 are differentiated from the cited reference for at least the same reasons as set forth above, as well as their own respective features.

The Examiner alternatively rejects claims 36-48 under 35 U.S.C. § 103(a) as being unpatentable over Schein in view of Owens et al., U.S. Patent No. 6,047,267 ("Owens"). Applicants respectfully traverse these rejections.

In general, Schein discloses a system and method for integrating data relating to customer transaction accounts based upon a customer's relationship with a financial institution. The Schein system logically links data from various accounts belonging to a customer to provide a more holistic view of the customer's relationship with the financial institution. Schein discloses a complex messaging system for managing data residing in geographically diverse locations, while ensuring that homogeneous data remains integrated. Schein further discloses "workflow data rules" that define how messages are to be routed.

The Examiner correctly notes that Schein, "does not use the words first, second and third high-level class" (page 13, paragraph 2). However, the Examiner next asserts that, "Owens discloses the use of relational databases in an object-oriented design in a multi-product on-line and Internet environment." Applicants respectfully disagree.

Owens discloses an object structure which allows a user to define new payment resources without requiring modifications to a relational database. An Owens object server automatically generates appropriate tables and columns for the relational database. When a new payment source is added to an account, a secondary object representing the payment source is created which inherits the properties of the container object.

It is important to note the differences between a true object database and a relational database incorporating an object programming paradigm for data management. Owens falls into the later category (namely, a relational database incorporating an object programming paradigm for data management), wherein objects represent a virtual view of a relational database structure, in that tables with complex linking structures can be represented within objects. As such, only the object needs to know the structure of the database. If a table within a database is modified, only objects referencing the table need to be modified. Therefore, functions and procedures in a program do not need to be modified, because they rely on the object to collect the required data. However, as pointed out by both Schein and Owens, such relational database architectures can result in slow searches. Therefore,

Owens employs objects to create the table structure and data in transient memory as an array; thus, the data can be searched in a more efficient manner. Moreover, the Owens objects may represent any number of configurations of linked tables. However, there is no disclosure of a one-to-one relationship between objects and specific products.

More significantly, while Owens incorporates class objects in order to model table structures and data outside of the database, there is no disclosure as to the specific relationship between the various class objects in order to efficiently maintain products. As such, neither Schein, Owens, nor any combination thereof, disclose or suggest at least a, "second subsection containing a high-level secondary class of objects and a second plurality of secondary classes of objects derived from said high-level secondary class of objects, wherein each of said second plurality of secondary classes of objects define one of said plurality of stored value products; and, wherein said second plurality of secondary classes of objects inherit attributes from said high-level key class of objects," as recited by independent claim 36.

Claims 37-48 depend from independent claim 36. Dependent claims 37-48 are differentiated from the cited references for at least the same reasons as set forth above, as well as their own respective features.

In view of the above remarks and amendments, Applicants respectfully submit that all pending claims properly set forth that which Applicants regard as their invention and are allowable over the cited references. Accordingly, Applicants respectfully request allowance of the pending claims. The Examiner is invited to telephone the undersigned at the Examiner's convenience, if that would help further prosecution of the subject Application. Applicants authorize and respectfully request that any fees due be charged to Deposit Account No. 19-2814, including any required extension fees.

Respectfully submitted,

Dated: September 27, 2006

By: _____
Howard Sobelman
Reg. No. 39,038

SNELL & WILMER L.L.P.
400 E. Van Buren
One Arizona Center
Phoenix, Arizona 85004
Phone: 602-382-6228
Fax: 602-382-6070
Email: hsobelman@swlaw.com